



CSM for SCCM Detection Rules

Last Modified on 31/08/2023 2:57 pm EEST

If you are uploading an application with custom detection method, you have multiple different options for the detection method. Detection method is currently in XML format, and you can set detection method with following types:

- Windows Installer
- Folder exists
- File
- File exists
- Registry key exists
- Registry value

You can define multiple detection rules by adding multiple detectionClause sections and also configure the comparison between rules. You need to include your detectionClause sections inside detectionMethods section, as shown in our examples.

Example with two different detection rules and their connector set as "and", meaning both rules must be met in order for SCCM client to detect the installation:

```
<detectionMethods>

  <detectionClause type="Directory" connector="and" Path="C:\Temp" DirectoryName="Folder 1"
Existence="$true"/>

  <detectionClause type="WindowsInstaller" connector="and" ExpectedValue="19.2.0"
ExpressionOperator="IsEquals" PropertyType="ProductVersion" ProductCode="{176E78BB-985C-427A-
AC57-A9AE850DDDFB}" />

</detectionMethods>
```

Windows installer

Modifiable values:

- **Connector**
 - If you have multiple detectionClause sections, you can configure the link between detection methods. If you have only one detectionClause section, this value can be either one.
- **ExpectedValue**
 - Value of the selected PropertyType. In general this is version of the windows installer.
- **ExpressionOperator**
 - Comparison type
- **PropertyType**
 - Property of the windows installer. Recommended to leave as ProductVersion
- **ProductCode**



- Product code of the windows installer<detectationMethods>

```
<detectionClause type="WindowsInstaller" connector="and,or" ExpectedValue="value"
ExpressionOperator="IsEquals,NotEquals,GreaterThan,LessThan,BeginsWith,EndsWith,Contains,NotContains"
PropertyType="ProductVersion" ProductCode="{productcode}" />

</detectationMethods>
```

Example 1 (Specific version)

```
<detectionMethods>

<detectionClause type="WindowsInstaller" connector="and" ExpectedValue="19.2.0"
ExpressionOperator="IsEquals" PropertyType="ProductVersion" ProductCode="{176E78BB-985C-427A-
AC57-A9AE850DDDFB}" />

</detectationMethods>
```

Example 2 (Any version)

```
<detectionMethods>

<detectionClause connector="and" type="WindowsInstaller" ProductCode="{111111-2222-4CCD-ABCD-
12345678}" Existence="$True" />

</detectationMethods>
```

Folder exists

Modifiable values:

- **Connector**
 - If you have multiple detectionClause sections, you can configure the link between detection methods. If you have only one detectionClause section, this value can be either one.
- **Path**
 - Path to a folder containing your target folder
- **DirectoryName**
 - Folder name

```
<detectionMethods>

<detectionClause type="Directory" connector="and,or" Path="c:\any\directory\path" DirectoryName="any
subdirectory under path" Existence="$true"/>

</detectationMethods>
```

Example:

```
<detectionMethods>

  <detectionClause type="Directory" connector="and" Path="C:\Program Files" DirectoryName="Centero
  Application" Existence="$true"/>

</detectionMethods>
```

File exists

Modifiable values:

- **Connector**
 - If you have multiple detectionClause sections, you can configure the link between detection methods. If you have only one detectionClause section, this value can be either one.
- **Path**
 - Path to a folder containing your target file
- **FileName**
 - File name
- **Is64Bit**
 - Add this parameter with \$true value to indicate that the target is 64 bit path
 - "This file or folder is associated with a 32-bit application on 64-bit systems." option is selected in detection method if you don't add this value at all

```
<detectionMethods>

  <detectionClause type="File" connector="and,or" Path="c:\any\directory\path" FileName="any filename
  under path" Existence="$true" Is64Bit="$true" />

</detectionMethods>
```

Example:

```
<detectionMethods>

  <detectionClause type="File" connector="and" Path="C:\Program Files\Centero Application"
  FileName="Application.exe" Existence="$true" Is64Bit="$true" />

</detectionMethods>
```



File attribute check

Modifiable values:

- **Connector**
 - If you have multiple detectionClause sections, you can configure the link between detection methods. If you have only one detectionClause section, this value can be either one.
- **Path**
 - Path to a folder containing your target file
- **FileName**
 - File name
- **ExpectedValue**
 - Value of the selected PropertyType. In general this is version of the windows installer.
- **ExpressionOperator**
 - Comparison type
- **PropertyType**
 - Property of the target file
- **Value**
 - Leave this as \$true
- **Is64Bit**
 - Add this parameter with \$true value to indicate that the target is 64 bit path
 - "This file or folder is associated with a 32-bit application on 64-bit systems." option is selected in detection method if you don't add this value at all

```
<detectionMethods>

  <detectionClause type="File" connector="and,or" Path="c:\any\directory\path" FileName="any filename
  under path" ExpectedValue="any value"
  ExpressionOperator="IsEquals,NotEquals,GreaterThan,GreaterEquals,LessThan,BeginsWith,EndsWith,Contains,NotCo
  PropertyType="Size,DateCreated,DateModified,Version" Value="$true" Is64Bit="$true" />

</detectionMethods>
```

Example:

```
<detectionMethods>

  <detectionClause type="File" connector="and" Path="C:\Program Files\Centero Application"
  FileName="Application.exe" ExpressionOperator="GreaterEquals" ExpectedValue="15.1.2.3"
  PropertyType="Version" Value="$true" Is64Bit="$true" />

</detectionMethods>
```



Registry key value

Modifiable values:

- **Connector**
 - If you have multiple detectionClause sections, you can configure the link between detection methods. If you have only one detectionClause section, this value can be either one.
- **Hive**
 - Registry hive
- **KeyName**
 - Path to a target key
- **ValueName**
 - Name of target value
- **ExpectedValue**
 - Target value
- **ExpressionOperator**
 - Comparison type
- **PropertyType**
 - Type of the target value
- **Value**
 - Leave this as \$true
- **Is64Bit**
 - Add this parameter with \$true value to indicate that the target is 64 bit path
 - "This registry key is associated with 32-bit application on 64-bit systems" option is selected in detection method if you don't add this value at all

```
<detectionMethods>

  <detectionClause type="RegistryKeyValue" connector="and,or"
Hive="LocalMachine,CurrentUser,CurrentConfig,ClassesRoot,Users" KeyName="any\key\path"
ValueName="any value name under key" ExpectedValue="any value"
ExpressionOperator="IsEquals,NotEquals,GreaterThan,LessThan,GreaterEquals"
PropertyType="string,integer,version" Value="$true" Is64Bit="$true" />

</detectionMethods>
```

Example:

```
<detectionMethods>

  <detectionClause type="RegistryKeyValue" connector="and" Hive="LocalMachine"
KeyName="SOFTWARE\Centero\Application" ValueName="Version" ExpectedValue="15.1.2.3"
ExpressionOperator="GreaterEquals" PropertyType="version" Value="$true" Is64Bit="$true" />
```

Registry key exists

Modifiable values:

- **Connector**
 - If you have multiple detectionClause sections, you can configure the link between detection methods. If you have only one detectionClause section, this value can be either one.
- **Hive**
 - Registry hive
- **KeyName**
 - Path to a target key
- **Is64Bit**
 - Add this parameter with \$true value to indicate that the target is 64 bit path
 - "This registry key is associated with 32-bit application on 64-bit systems" option is selected in detection method if you don't add this value at all

```
<detectionMethods>

  <detectionClause type="RegistryKey" connector="and,or" Is64Bit="$true"
Hive="LocalMachine,CurrentUser,CurrentConfig,ClassesRoot,Users" KeyName="any\key\path"/>

</detectionMethods>
```

Example:

```
<detectionMethods>

  <detectionClause type="RegistryKey" connector="and,or" Is64Bit="$true"
Hive="LocalMachine,CurrentUser,CurrentConfig,ClassesRoot,Users" KeyName="any\key\path"/>

</detectionMethods>
```